REMARKS

Claim 1-37 are pending in the application and stand rejected. Claims 1-12, 18-20, 22-29, and 31-37 have been amended. Claims 21 and 30 have been canceled without prejudice. New claims 38-42 have been added. No new matter has been introduced by virtue of the claim amendments.

The claims have been amended in a sincere effort to further clarify the patentable distinctions of the claimed invention over the cited art of record, as well as resolve what appears to be a continued confusion and misinterpretation as to the scope and meaning of the claim features "scheduled activities" and determining "a recommended order for scheduled activities", leading to the Examiner's finding of a purported "inconsistency" between "scheduled activities" and determining a recommended order in which the scheduled activities can be enacted.

By way of specific example, the Examiner asks on page 4, paragraph 5, of the Office Action: **"If the activities are scheduled (i.e., there timing and therefore the order they occur) when how can these same activities then have a recommended order determined for them?"** Moreover, in finding that claims 1-37 are indefinite under 35 USC 112, second paragraph, the Examiner asserts that:

> **Scheduling an activity implies that a time is set for the activity (i.e., a schedule of activities includes when those activities will occur, e.g., a schedule of classes, a schedule of entertainment events for a holiday celebration). Thus, having scheduled activities and then determining that there is a recommended order to those activities then implies that those activities were not in fact, "scheduled".**

Generally speaking, a fundamental flaw in the above reasoning stems from an misconstruction of the terms, e.g., "scheduled activity" and "scheduling an activity", which is seemingly based on a vacuum interpretation that fails to consider the teachings in Applicants' specification coupled with an unreasonable parsing of the claim terms where the claim terms are construed in isolation rather than in the context of the claimed inventions as a whole.

By way of example, there is seemingly no basis for construing the claim term "scheduling activities" as implying that the timing and **order at which the activities occur** are also specified by virtue of being scheduled. The Examiner has pointed to nothing in Applicants' specification that even remotely supports this claim construction, and such claim construction is contrary to the teachings in Applicants' specification. Indeed, within the context of the claimed inventions, a "scheduled activity" may properly refer to an activity that has been initiated for execution at a given time when the schedule rule of the activity is evaluated to be true. For example, paragraph [0053] of Applicants' Published Application (US 2003/0018508) states as follows:

> **I(B)(ii). Schedule Rule: A schedule rule according to the present invention comprises a Boolean expression over the activity inputs that specifies the conditions under which the data-triggered workflow engine should schedule the activity for enactment. The difference between the permitted rule and the schedule rule is that the permitted rule is checked when the participant takes the initiative to do an activity, whether or not it is on a worklist. The schedule rule directs the data-triggered workflow engine to take the initiative to inform the participant that the participant should perform an activity (typically by placing the activity on the worklist).**

The above disclosure, as well as other relevant disclosure in Applicants' specification, clearly teaches that a "scheduled activity" is an activity that is initiated for execution (e.g., placed on the worklist) when the conditions specified by the schedule rule of the activity are met, and that that execution of the scheduled activity is at the option of the workflow participant. There is nothing in the above disclosure or other sections of Applicants' disclosure that supports the Examiner's interpretation that a "scheduled activity" is an activity that is scheduled at a given time and order for execution.

Moreover, in the context of the claimed inventions as a whole, the impropriety of the interpretation of the term "scheduled activity" is underscored by the fact that such interpretation results in a finding of inconsistency with the claim term *"computing a recommended order in which the scheduled activities can be enacted . . .".* In the Office Action, the Examiner asks: **If the activities are scheduled (i.e., there timing and therefore the order they occur) when how can these same activities then have a recommended order determined for them?**, The answer is simple – scheduling activities within the context of the claimed inventions <u>does not imply that timing and order of execution is specified,</u> but merely refers to the activities being initiated for execution (scheduled activities), whereby "scheduling" and "determining a recommended order of execution" are two distinct mechanisms of the claimed inventions. This is fully supported and clearly explained in Applicants' specification. For example, as stated in paragraph [0025] of the Application Publication:

> **The present invention is directed to a system and method for providing data-triggered workflow management. In one aspect, the invention defines a data-triggered process definition language. Each activity specified in a preferred process definition language is permitted to be enacted whenever a specified combination of data conditions (the *permitted* rule) is met, regardless of which activities have previously been enacted. An activity is scheduled for enactment when a stricter combination of data conditions become true (the *schedule* rule). A data-triggered workflow engine comprises an activity scheduler that utilizes the current state of a process instance, the permitted and schedule rules, an activity network, and additional attributes of activities to schedule the enactment of activities. <u>In contrast to conventional process definition languages, however, an activity network does not completely prescribe the enactment order, but rather controls what enactment order the data-triggered workflow engine will *suggest* to a participant. The activity scheduler computes attributes of activities that suggest an order in which to enact the activities, based on the information it has. A participant, however, may select a different order based on other information, and can even enact activities that have not been scheduled.</u> The activity scheduler does not assume that the suggested order has been followed. It simply responds to each activity enactment event (e.g. start, finish, cancel) by revising the suggested enactment order.** (emphasis added).

Accordingly, it is respectfully submitted that the Examiner has incorrectly characterized and improperly interpreted claims terms regarding *scheduled activities* and *determining order of enacting scheduled activities,* as contemplated by the claimed inventions.

Claim Rejections – 35 U.S.C. 112.

Claims 1-37 are rejected as being indefinite for the reasons set forth on pages 4-5 of the Office Action. Applicants respectfully disagrees with these rejections for at least those reasons explained above, in that the purported claim term "inconsistencies" are based on the Examiner's mischaracterization of the claimed inventions.

In any event, as noted above, the claims have been amended in a sincere effort to further clarify the claimed inventions. For instance, the inventions of claims 1, 20 and 29 commonly claim, for example, *wherein the process definition defines activities that are associated with the workflow process, and wherein the process definition includes an activity specification for each activity associated with the workflow process, wherein the activity specification for each activity includes a schedule rule that specifies one or more conditions under which the activity is initiated for execution based on workflow relevant data, independent of control flow dependencies.*

These claim amendments at the very least render the 112 rejections moot, and as described below, clearly distinguish the claimed inventions over the cited art of record.

Claim Rejections – 35 U.S.C. 103

Claims 1-37 are rejected as being unpatentable over Hollingsworth and the Workflow Management Coalition in view of U.S. Patent Application No. 2002/0055849 by Georgakopoulos. Applicant respectfully submits that at the very least, the cited combination of references does not disclose or suggest various elements of claims 1, 20 and 29.

For instance, with regard to claims 1, 20 and 29, the cited combination of references does not disclose or suggest *an activity specification for each activity includes a schedule rule that specifies one or more conditions under which the activity is initiated for execution based on workflow relevant data, independent of control flow dependencies*. The use of "schedule-rules" is in contrast to conventional schemes in which rules for enacting activities for execution are derived from e.g., "control flow dependencies" in state-based workflow systems such as in disclosed in Hollingsworth and Georgakopoulos.

In conventional workflow systems, *control flow dependencies* are specified to impose restrictions on the occurrence and order of the activity instances within a process. C*ontrol flow dependencies* include control flow primitives, or transitions, which specify the control flow transitions from a source to target activity based on a transition rules (e.g., upon completion of source activity*)*. *Control flow d*ependencies may also include transition conditions that are attached to a transition to control the validity of the transition, or otherwise specify criteria for state transitioning from the source (current activity) to the target (next) activity. In short, control flow dependencies are transition conditions that are evaluated to control the transition from a source activity to a target activity (i.e., transition to a target activity that is initiated for execution when the transition conditions are met). (see, generally, Georgakopoulos, paragraphs [007], -[009]; Hollingsworth, pp. 23-24 (discussion of activity state transitions) and p. 54, definition of transition condition, etc.)

In contrast to the conventional schemes in which the *control flow dependencies* are used to determine when to initiate an activity for execution, the schedule rule for a given activity specifies one or more conditions under which the activity is initiated for execution based on workflow relevant data, *independent of control flow dependencies*. In other words, the claimed "schedule rules" provide a mechanism to determine when an activity can be initiated for execution independent of transition condition (e.g., independent of the completion of preceding activities) and regardless of whether or not the given activity is the target of any control flow transition.

Neither Hollingsworth nor Georgakopoulos disclose or suggest "schedule rules" as contemplated by the claimed inventions. The Examiner relies on page 13 of Hollingsworth as teaching "schedule rules" based on the general statement that the *workflow enactment software interprets the process description and controls the instantiation of processes and sequencing of activities, adding work items to the user work lists ....* etc. However, the cited section undoubtedly does not disclose or remotely suggest *schedule rules" which specify the conditions over workflow relevant data, which are to be satisfied for a given activity to be initiated for execution, independent of control flow dependencies,* as claimed.

In fact, Hollingsworth teaches (on pages 23 and 24) that the workflow enactment service is a "state transition machine" where individual process or activity instances change states in response to external events (completion of activity) or control decisions taken by a workflow engine. In this regard, Hollingsworth teaches (on pages 23 and 24) nothing more than a conventional workflow enactment process discussed above whereby activities are initiated for execution based on activity state transitions. Hollingsworth does not disclose any mechanism for initiating activities for execution based on schedule rule conditions, independent from control flow dependencies.

Moreover, Georgakopoulos does not disclose or suggest *schedule rules are used to specify one or more conditions under which one or more activities can be initiated for execution based on workflow relevant data, independent of control flow dependencies.* Georgakopoulos discloses the use of "inhibitor dependencies" and "option dependencies", for example, as control flow primitives providing extensions to conventional control flow dependencies for purposes of flow control [see, e.g., Georgakopoulos paragraphs [0010] and [0011]. As explained hereafter, inhibitor dependencies" and "option dependencies", are clearly distinguishable from the claimed "schedules rules".

Georgakopoulos teaches that inhibitor primitives provide execution control for processes having mutually exclusive execution interdependencies, where an inhibitor primitive creates an inhibitor dependency that prevents a target activity from starting after a sourcing activity has started. For instance, in the example of "inhibitor dependencies" provided in FIGs. 2A and 2B, Georgakopoulos teaches a process definition in which Activity S transitions to both Activities A and B (via transitions (201)), i.e., Activities A and B are enabled once activity S has been performed, and where inhibitor dependencies (200a) and (200b) are attached to the Activities A and B to implement a flow control requirement that either A or B but not both could be executed. Georgakopoulos teaches that when either activity A or activity B is selected and starts to execute, the appropriate inhibitor dependency (200) is activated to inhibit the instantiation of the other non-selected activity [see paragraphs 0035 and 0036].

In this regard, the inhibitor dependencies are clearly different from the claimed "schedule rules", as the inhibitor dependencies do not serve as conditions for a given activity under which the activity is initiated for execution based on workflow relevant data, *independent of control flow dependencies*. Indeed, the inhibitor dependencies for a given activity are conditions that are evaluated when the associated activity actually starts to execute for purposes of disabling another initiated (target) activity from being executed.

Georgakopoulos further teaches option primitives which are repeatable creator primitive that permit an activity enabled by a normal control flow to be instantiated zero or more times. In particular, Georgakopoulos illustrates an example of the "option primitive" in FIGs. 3A and 3B, whereby after activity S is completes and Activity B is enabled by virtue of the execution control flow transition (304), the option primitive (312) attached to activity B allows the enabled activity B to be instantiated zero or more times (see paragraphs 0039-0041).

In this regard, the option primitive is clearly different from the claimed "schedule rules", as the option dependencies do not serve as conditions for a given activity under which the activity is initiated for execution based on workflow relevant data, *independent of control flow dependencies*. Indeed, the option primitives for a given activity are conditions that are evaluated only when the given activity is actually initiated for execution via the control flow dependencies (e.g., transition (304)).

Moreover, the teachings of Georgakopoulos with regard to placeholder primitives (or abstract activity primitive) do not teach the claimed schedule rules for similar reasons discussed above. Indeed, Georgakopoulos teaches that activity placeholders are enabled by control flow, and allow late binding of activities to a running workflow (see, paragraph {0016]). Placeholder primitives are not conditions that serve to initiate the execution of activities independent of control flow dependencies.

In short, Applicant respectfully contends that Hollingsworth and Georgakopoulos, either singularly or in combination, fail to disclose or suggest *schedule rules that specify conditions under which activities are initiated for execution independent of control flow dependencies*, as essentially recited in claims 1, 20 and 29. Indeed, as explained above, Hollingsworth and Georgakopoulos do not disclose process definitions providing specific mechanism such as schedule rules that allows an activity to be initiated for execution (scheduled) independent of control flow dependencies (e.g., independent of the completion of a preceding activity). Again, the claimed "schedule rules" allow a given activity to be initiated for execution regardless of whether or not the given activity is the target of any control flow transition.

For at least the above reasons, claims 1, 20 and 29 are patentable over the cited combination of references. Moreover, all pending dependent claims are patentable over the cited combination of references at least by virtue of their dependence from respective base claims 1, 20 or 29. In any event, Applicant respectfully contends that the dependent claims are patentable over the cited reference in their own right.

For instance, with regard to claims 38, 39 and 41, the cited combination of references do not disclose or suggest *an activity network specification comprising one or more relations between activities which are used to determine a suggested sequence of executing two or more activities that are initiated for execution at a given time*, whereby when two or more activities are initiated for execution at a given time, the one or more relations between the two or more initiated activities *are evaluated based on the execution state of the process instance to determine a suggested sequence for executing the two or more initiated activities*, and whereby the suggested execution sequence is at the option of the workflow participant.

In the Office Action, the Examiner relies on paragraphs 13 and 14 of Georgakopoulos as teaching a process of *recommending an order in which scheduled activities can be enacted.* Although not clear, the Examiner's reliance in this regard is seemingly based on the "option primitive" control flow mechanism disclosed by Georgakopoulos as discussed above. Applicant respectfully disagrees with the Examiner's characterization of the "option primitives" of

Georgakopoulos as teaching *recommending an order in which scheduled activities can be enacted,* for those reasons explained in previous responses.

In any event, it is fundamentally clear that "option primitive" control flow mechanism disclosed by Georgakopoulos does not disclose or suggest *activity network specification comprising one or more relations between activities which are used to determine a suggested sequence of executing two or more activities that are initiated for execution at a given time,* as essentially claimed in claims 28, 29 and 41. Indeed, as discussed above, Georgakopoulos discloses in FIGs. 2A/2b (inhibitor primitives) and 3A/3B (option primitives), examples of control flow where two activities A and B are initiated for execution a given point in the process flow after completion of activity S (i.e., activities A and B are enabled for execution in a conventional manner based on evaluation of control flow dependencies controlling the transition from S to A and B).

However, in both examples of FIGs. 2 and 3, although two or more activities are concurrently initiated for execution via standard control flow transitions, there is simply nothing in Georgakopoulos that discloses or suggests a mechanism *to determine a suggested sequence of executing two or more activities that are initiated for execution at a given time,* as claimed. Indeed, with regard to FIG. 2, Georgakopoulos explicitly teaches that the decision to select activity A or B is made by the participant (see, paragraph [0036]. There is no teaching that the system provides any suggestion or recommendation with regard to selecting activity A or B for execution, much less a suggested sequence for executing the initiated activities A and B. Similarly, with regard to FIG. 3, after S completes, a control flow transition (304) operates to enable Activity B and some other activity at the same time (see paragraph [0041]. However, there is no suggestion or recommendation given to a participant regarding execution of sequence of two or more enabled activities, much less a recommendation for selecting one of the enabled activities.

Furthermore, although Georgakopoulos discloses in paragraph [0014] that the option and inhibitor primitives permit the workflow system to "suggest" one or more optional activities to its participants at a specific point in the workflow process execution, when viewed in the context of the overall teachings of Georgakopoulos, it is clear that the term "suggest" merely refers to the process by which several Activities can be initiated for execution at some prespecified point in the control flow via control flow transitions, allowing a participant to select a desired activity for execution. However, there is a stark difference between the Georgakopoulos process by which several Activities initiated for execution by control flow transitions are merely made available for selection by the user, as opposed to a process for evaluating relations between two or more activities that are initiated for execution at a given time to *determine a suggested sequence of executing two or more activities,* as claimed.

For at least the above reasons, withdrawal of the claim rejection is requested.

Respectfully submitted,

Donald B. Paschburg

Donald B. Paschburg
Reg. No. 33,753
Attorney for Applicant

Siemens Corporation
Intellectual Property Department
170 Wood Avenue South
Iselin, New Jersey 08830
(732) 321-3100